

# Banco de Dados

Prof. Willian Douglas Ferrari Mendonça  
e-mail: [williandouglasferrari@gmail.com](mailto:williandouglasferrari@gmail.com)

# **História e Introdução SQL**

# Introdução SQL

- ❖ SQL pode ser considerada um dos principais motivos para o sucesso dos bancos de dados relacionais comerciais.
- ❖ Com ela se tornou um padrão para esse tipo de banco de dados
- ❖ Os usuários ficaram menos preocupados com a migração de suas aplicações de outros tipos de sistemas de dados
- ❖ Na prática, é óbvio, existem muitas diferenças entre diversos pacotes de SGBD.

# Introdução

- ❖ A linguagem SQL oferece uma interface de linguagem declarativa de nível mais alto.
- ❖ Modo que o usuário apenas especifica qual deve ser o resultado.
- ❖ Deixando a otimização real e as decisões sobre como executar a consulta para o SGBD.
- ❖ Embora a SQL inclua alguns recursos da álgebra relacional, ela é baseada em grande parte no cálculo relacional de tupla.

# História

- ❖ SQL hoje é expandido como Structured Query Language( Linguagem de Consulta Estruturada)
- ❖ Originalmente SQL era chamada de SEQUEL(Structured English Query Language)
- ❖ Foi criada pela IBM como a interface para um sistema de banco de dados relacional experimental chamado SYSTEM R.A

# História

- ❖ Um esforço conjunto entre a ANSI(American National Standards Institute) e a ISO(Internacional Standards Institute) levou a versão padrão da SQL (ANSI, 1986) chamada SQL-86 ou SQL1
- ❖ Um padrão revisado e bastante expandido SQL-92 desenvolvido um pouco mais tarde
- ❖ O próximo padrão reconhecido foi SQL:1999 que começou com SQL3
- ❖ Duas outras atualizações SQL:2003 e SQL:2006 acrescentaram recursos XML.

# História

- ❖ SQL é uma linguagem de banco de dados abrangente
- ❖ Tem instruções para definição de dados, consultas e atualizações
- ❖ Além disso, ela tem facilidades para definir visões sobre o banco de dados, para especificar segurança e autorização
- ❖ Ela também possui regras para embutir instruções SQL em uma linguagem de programação de uso geral, como C/C++ , JAVA, C# e etc....



# **Definições e tipos de dados em SQL**



# SQL Básica

- ❖ DDL = CREATE, ALTER e DROP.
- ❖ DML = SELECT, INSERT, UPDATE e DELETE.
- ❖ DCL = GRANT e REVOKE.
- ❖ TC = COMMIT, ROLLBACK e SAVEPOINT.

# SQL Básica

- ❖ Definição de dados -> Comandos DDL permite a criação e organização dos dados;
- ❖ Manipulação de dados -> Comandos DML possibilitam a manipulação dos dados armazenados;
- ❖ Controle de acesso -> Proteção contra manipulação;
- ❖ Integridade dos dados -> Define regras de integridade;
- ❖ Independência e portabilidade de fabricante -> Incorporada em vários SGBD, para ser usado em computadores até celulares;

# Tabela

- ❖ São as estruturas mais importantes em um banco de dados. Nas tabelas estará todo o conteúdo que representa cada objeto no mundo real (entidades).
- ❖ Uma tabela é composta de colunas (atributos de uma entidade), que por sua vez formam os registros (tuplas).

# Constraints

- ❖ Constraints são integridades a serem definidas em tabelas. Pode-se definir integridades para:
- ❖ Definir campos obrigatórios;
- ❖ Chaves primárias;
- ❖ Chaves estrangeiras.

# Constraints

- ❖ CHAVE PRIMÁRIA
  - ❖ Coluna identificadora de um registro em uma tabela, para representa-la basta acrescentar a chave primary key (coluna).
- ❖ CHAVES ESTRANGEIRA
  - ❖ Campo que estabelece o relacionamento entre duas tabelas, para representação é utilizado a chave foreign key (coluna) e references (tabela).

# Constraints

## Exemplo

```
CREATE TABLE cidade  
(codigoCidade integer NOT NULL,  
descricao varchar(40) NOT NULL,  
estado char(2),  
PRIMARY KEY (codigoCidade))
```

```
CREATE TABLE cliente  
(codigoCliente integer NOT NULL,  
nome varchar(30) NOT NULL,  
codCidade integer,  
PRIMARY KEY (codigoCliente),  
FOREIGN KEY (codCidade) REFERENCES Cidade (codigoCidade))
```

Cliente

codigoCliente	Nome	endereço	codigoCidade
548	Maria	Rua Carvalho 615	1
549	Pedro	Rua Pedro Chaves 22	5

Viola a restrição → cidade 5 não existe

Cidade

codigoCidade	Descricao	Estado
1	Florianópolis	SC
2	São José	SC

A restrição garante que não irá existir CLIENTE que more numa cidade que não exista na tabela CIDADE



# Constraints

- ❖ NOT NULL Indica que o conteúdo de uma coluna da tabela não poderá ser nulo.
- ❖ DEFAULT Define o conteúdo padrão a uma coluna de uma tabela, através da representação default valor.



# Constraints

- ❖ **UNIQUE** Um atributo unique é aquele que é considerado único em uma tabela, isto é, valores dessa coluna não podem ser repetidos.
- ❖ A chave primária é um exemplo de um atributo com a regra unique. A declaração de um atributo unique é realizada na criação da tabela.

# Constraints

- ❖ CHECK A regra check define o intervalo de dados que pode ser usado nos campos de uma coluna.
- ❖ No exemplo só podem ser aceitos pesos maiores que 0.

```
create table pessoa(  
cod_pes integer not null check (cod_pes>0),  
nome_pes varchar(255) not null );
```

# SQL Básica

- ❖ A SQL usa os termos tabela, linha e coluna para os termos do modelo relacional formal relação, tupla e atributo
- ❖ O principal comando SQL para definição de dados é o CREATE
- ❖ Que pode ser usado para criar esquemas, tabelas e domínios

# SQL Básica

- ❖ Conceitos de esquema e catálogo em SQL:
  - ❖ As primeiras versões do SQL não incluíam o conceito de esquema.
  - ❖ Todas as relações faziam parte do mesmo esquema.
  - ❖ Foi incorporado com SQL2.
  - ❖ A fim de agrupar tabelas e outras construções de bancos de dados.
- ❖ Um esquema/banco de dados SQL
  - ❖ É identificado por um nome
  - ❖ E pode incluir um identificador de autorização
  - ❖ O esquema é criado por meio da instrução CREATE SCHEMA
- ❖ **Create schema** empresa;
- ❖ **Create database** empresa;

# SQL Básica

- ❖ Comando CREATE TABLE em SQL
  - ❖ O comando CREATE TABLE é usado para especificar uma nova relação, dando-lhe um nome e especificando seus atributos e restrições iniciais
  - ❖ Os atributos são especificados primeiro:
    - ❖ Cada um deles recebe um nome, um tipo de dado para especificar seu domínio de valores e quaisquer restrições de atributo como NOT NULL
    - ❖ As restrições de chave, integridade de entidade e integridade referencial podem ser especificadas na instrução CREATE TABLE.

# Tipos de dados SQL Básica

Os tipos de dados básicos disponíveis para atributos são numérico, cadeia ou sequência de caracteres, cadeia ou sequência de bits, booleano, data e hora.



# Tipos de dados SQL Básica

- ❖ Tipos de dados numérico:
  - ❖ Números inteiros de vários tamanhos INTEGER OU INT E SMALLINT
  - ❖ Números de pontos flutuante FLOAT e REAL E DOUBLE PRECISION
  - ❖ Formato DECIMAL(i, j) ou DEC(i, j) ou NUMERIC(i, j), onde i é o número de dígitos decimais e j é número de dígitos após o ponto decimal



# Tipos de dados SQL Básica

- ❖ Tipos de dados de cadeia de caracteres:
  - ❖ São de tamanho fixo CHAR(n) onde n número de caracteres ou tamanho variável VARCHAR(n) onde n é o número máximo de caracteres.
  - ❖ Para valores maiores existe o CLOB, pode ser especificado em kilobytes(K), megabytes(M) ou gigabeytes(G)

# Tipos de dados SQL Básica

- ❖ Tipos de dados cadeia de bits:
  - ❖ Podem ser tamanho fixo n BIT(n) ou tamanho variável BIT VARYING(n), onde n é o número máximo de bits.
  - ❖ Outro tipo de cadeia de bits de tamanho variável BINARY LARGE OBJECT ou BLOB, como o CLOB tamanho pode variar de K à G exemplo: BLOB(30G)

# Tipos de dados SQL Básica

- ❖ Tipos de dado booleano:
  - ❖ Tem os valores tradicionais Verdadeiro(TRUE) e Falso(FALSE)
  - ❖ No SQL existe um terceiro valor o indefinido(UNKNOWN)

# Tipos de dados SQL Básica

- ❖ Tipos de dado DATE:
  - ❖ Possui dez posições, e seus componentes são DD-MM-YYYY
  - ❖ TIME(tempo) pelo menos 8 posições HH:MM:SS
- ❖ Timestamp esse tipo de dado inclui DATE e TIME mais 6 posições para frações decimais de segundos

# Tipos de dados SQL Básica

- ❖ É possível especificar tipo de dado de cada atributo diretamente.
- ❖ Como alternativa um domínio pode ser declarado e seu nome usado como a especificação de atributo.
- ❖ Isso torna mais fácil mudar o tipo de dado para um domínio que é usado por diversos atributos
- ❖ Exemplo:
  - ❖ `CREATE DOMAIN TIPO_CPF AS CHAR(11)`